6. Statistical Tests and Confidence Intervals

One Mean or the Difference of Two Means

`out = t.test(x, y = NULL, alternative = "two.sided", mu = 0, conf.level = .95)` tests $H_0 : \mu_X = \mu_0 =$ `mu` for a sample `x` from a normal population; or, if `y` is given, $H_0 : \mu_X - \mu_Y = \mu_0 =$ `mu`, for samples `x` and `y` from normal populations. `out` is a list containing (among other things):

- `$parameter`: degrees of freedom ($n - 1$, where $n =$ `length(x)`, if `y == NULL`; or a mess)
- `$statistic`: Student's t test statistic, $t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$; or $t = \frac{(\bar{x} - \bar{y}) - \mu_0}{\sqrt{\frac{s_X^2}{n_X} + \frac{s_Y^2}{n_Y}}}$
- `$p.value`: probability of a value at least as extreme as t under H_0
- `$conf.int`: confidence interval for μ_X (or $\mu_X - \mu_Y$) corresponding to H_1 in `alternative`
- `$estimate`: $\bar{x}$ (or $\bar{x}$ and $\bar{y}$)

Other `alternative` choices are `"less"` and `"greater"`. e.g.

```
x = rnorm(n = 10, mean = 0, sd = 1); (out = t.test(x))
x = rnorm(10, 0, 1); (out = t.test(x, mu = 2))  # rnorm() isn't part of the test!
x = rnorm(10, 0, 1); y = rnorm(10, 2, 1); (out = t.test(x, y))|
x = rnorm(10, 0, 1); y = rnorm(10, 2, 1); (out = t.test(x, y, mu = -2))
```

F Test for Equality of Variances

`out = var.test(x, y, ratio = 1, alternative = "two.sided", conf.level = .95)` tests $H_0 : \frac{\sigma_X^2}{\sigma_Y^2} =$ `ratio` for two samples `x` and `y` from normal populations. `out` is a list containing:

- `$parameter`: degrees of freedom ($n_X - 1$ and $n_Y - 1$, where $n_X =$ `length(x)` and $n_Y =$ `length(y)`)
- `$statistic`: F test statistic, $f = \frac{s_X^2}{s_Y^2}$
- `$p.value`: probability of a value at least as extreme as f under H_0
- `$conf.int`: confidence interval for $\frac{\sigma_X^2}{\sigma_Y^2}$
- `$estimate`: $\frac{s_X^2}{s_Y^2}$

e.g. `x = rnorm(100, 0, 1); y = rnorm(10, 0, 2); (out = var.test(x, y, ratio = 1))`

e.g. `x = rnorm(100, 0, 1); y = rnorm(10, 0, 2); (out = var.test(x, y, ratio = .25))`

Chi-Squared Tests

- Goodness-of-fit:

  `counts = c(...); probs = c(...); (out = chisq.test(x = counts, p = probs))`
  tests H_0 : "`counts` came from a distribution with probabilities `probs`". e.g.

  `counts=c(12,15,17,6); probs=c(.20,.25,.40,.15); (out=chisq.test(x=counts, p=probs))`

  `out` is a list containing (among other things):

  - `$parameter`: degrees of freedom (#categories − 1 == `length(x) - 1`)
  - `$statistic`: χ^2 test statistic for goodness-of-fit of observed `counts` to proposed `probs`
  - `$p.value`: probability of a value at least as extreme as χ^2 under H_0

- Independence / Homogeneity

  e.g. Consider the counts in this contingency table:

	Smoking status			
Education	Nonsmoker	Former	Moderate	Heavy
Primary	56	54	41	36
Secondary	37	43	27	32
University	53	28	36	16

  To get data into the test, we need x = `matrix(data, nrow, ncol, byrow = FALSE)`, which fills an `nrow` by `ncol` matrix `x`, by column, from the vector `data`. Note that `x[,c]` is the c^{th} column of `x`, and `x[r,]` is the r^{th} row. e.g.

  `(x = matrix(data = c(56,37,53, 54,43,28, 41,27,36, 36,32,16), nrow=3, ncol=4))`

  The χ^2 test `out = chisq.test(x)` is for H_0 : "row and column variables are independent" (or H_0 : "the column populations have the same distribution with respect to the row variable").

  `out` is a list containing (among other things):

  - `$parameter`: degrees of freedom, (#rows − 1) × (#columns − 1)
  - `$statistic`: χ^2 test statistic for independence of row and column variables (or for homogeneity of column populations with respect to row variable)
  - `$p.value`: probability of a value at least as extreme as χ^2 under H_0
  - `$expected`: expected counts under H_0

  To use `chisq.test()` on variables in a data frame, recall that `table(...)` makes a contingency table of counts of each combination of factors in `...`. e.g.

```
table(mtcars$cyl)
table(mtcars$cyl, mtcars$gear)
```

One Proportion or the Difference of Two Proportions

- For integers `x` and `n`, `out = prop.test(x, n, p, alternative = "two.sided", conf.level = .95)` tests $H_0 : p = p_0 =$ `p` for a sample containing `x` successes in `n` trials. e.g.

  `x = 800; n = 1000; p0 = .77; (out = prop.test(x, n, p0, correct=FALSE))`

  (`correct=FALSE` disables a good continuity correction that would add to the explanation.)

  `out` is a list containing (among other things):

  - `$parameter`: degrees of freedom (#categories $- 1 = 2$ (success and failure) $- 1 = 1$)
  - `$statistic`: χ^2 test statistic for goodness-of-fit of observed counts, `x` successes (800) and `n-x` failures ($1000 - 800 = 200$), to the distribution with expected counts, `n*p` successes ($770 = .77 \times 1000$) and `n*(1-p)` failures ($230 = (1 - .77) \times 1000$).
  - `$p.value`: probability of a value at least as extreme as χ^2 under H_0
  - `$conf.int`: confidence interval for p
  - `$estimate`: $\hat{p} =$ `x/n`

  (I teach an equivalent z-test for this one-proportion test:

  `phat = x/n; z = (phat - p0) / sqrt(p0*(1-p0)/n); (p.value = 2*pnorm(-abs(z)))`

  Then `z^2` matches `out$statistic` above, and the P-values are the same.

)

- For 2-vectors `x` and `n`, `out = prop.test(x, n, alternative = "two.sided", conf.level = .95)` tests $H_0 : p_1 - p_2 = 0$ (or $p_1 = p_2$) for samples from two populations containing `x[1]` successes in `n[1]` trials and `x[2]` successes in `n[2]` trials, respectively. e.g.

  `x = c(40, 87); n = c(244, 245); (out = prop.test(x, n, correct=FALSE))`

  `out` is a list containing (among other things):

  - `$parameter`: degrees of freedom, 4 (counts) $-$ 2 (constraints due to the sample sizes) $-$ 1 (parameter estimated, $\hat{p}$) $= 1$

	Sample	
Outcome	1	2
success	40	87
failure	$244 - 40$	$245 - 87$

  - `$statistic`: χ^2 test statistic for goodness-of-fit of observed counts, `x[1]` and `x[2]` successes (40 and 87) and `n[1]-x[1]` and `n[2]-x[2]` failures ($244 - 40$ and $245 - 87$) to the distribution with corresponding expected counts based on $\hat{p} = \frac{x_1+x_2}{n_1+n_2}$
  - `$p.value`: probability of a value at least as extreme as χ^2 under H_0
  - `$conf.int`: confidence interval for the difference in proportions $p_1 - p_2$
  - `$estimate`: a 2-vector containing $\hat{p}_1 =$ `x[1]/n[1]` and $\hat{p}_2 =$ `x[2]/n[2]`

  (Here, too, I teach an equivalent z-test, with `z^2 == out$statistic`, and the same P-value.)